End – SEMESTER EXAMINATION
UG

Session: July-December, 2024 Semester: 7th Time: 3 Hrs. Full Marks: 100

Course Code: UCSE701 Course Title: Advance Algorithms

ANSWER All QUESTIONS

1. (i) When using the in-place Quicksort algorithm, which of the following operations has a significant impact on the algorithm's efficiency?

a) The choice of sorting order (ascending or descending)
b) The number of swaps made during the partitioning process.
c) The number of recursive calls made by the algorithm.
d) The initial order of the elements in the array.

(ii) In the worst case, Quicksort has a time complexity of $O(n^2)$. Which of the following scenarios will cause Quicksort to achieve its worst-case time complexity?

a) The pivot is always the median of the array
b) The pivot is always the smallest or largest element in the array
c) The pivot is chosen randomly each time
d) The array is already sorted

(iii) Which of the following best describes amortized analysis?

a) It calculates the worst-case time complexity for each operation.
b) It averages the worst-case time complexity across all operations.
c) It calculates the best-case time complexity for each operation.
d) It averages the time complexity over a sequence of operations.

(iv). For a data structure where each operation has a worst-case cost of O(nlogn), but the amortized cost of a sequence of operations is O(1), which of the following data structures could this describe?

a) Binary Search Tree (BST) b) Dynamic Array
c) AVL Tree d) Hash Table with chaining

(v) Which of the following algorithms is the most efficient choice for finding the maximum flow in a network with a small number of edges but large capacity values?

a) Ford-Fulkerson Algorithm b) Edmonds-Karp Algorithm
c) Dijkstra's Algorithm d) Prim's Algorithm

(vi) In a 2-approximation algorithm for the Vertex Cover problem, what is the maximum ratio between the cost of the approximate solution and the cost of the optimal solution?

a) 1 b) 1.5
c) 2 d) 3

(vii) If problem A is NP-Hard, which of the following is correct regarding problem B if A is reducible to B in polynomial time?

a) Problem B is NP. b) Problem B is NP-Complete.
c) Problem B is NP-Hard. d) Problem B cannot be NP-Complete.

(viii) If it is proven that P = NP, what would be the consequence for NP-Hard problems?

a) They would become solvable in exponential time.
b) They would remain unsolvable in polynomial time.
c) They would all be solvable in polynomial time.
d) They would be unsolvable.

(ix) If a problem is in the NP class, which of the following must be true?
a) It can be solved in polynomial time. b) It can be verified in polynomial time.
c) It is NP-Hard. d) It is both NP and NP-Hard.

(x) Which of the following best describes the Max-Flow Min-Cut Theorem?
a) The maximum flow in a network is always less than the minimum cut.
b) The minimum cut in a network is equal to the maximum flow through the network.
c) The maximum flow in a network is always greater than the minimum cut.
d) There is no relationship between maximum flow and minimum cut.

(xi) Consider a stack that supports a new operation, PopMany, which pops multiple elements at once. If the amortized cost of a single Push operation is 1 and the amortized cost of PopMany is proportional to the number of elements popped, find the worst-case amortized cost of a sequence of operations that includes multiple Push and PopMany operations?

(xii) You are implementing a dynamic array (like a list in Python or a vector in C++). When the array reaches its capacity, it doubles its size. Find is the amortized time complexity of inserting an element into this dynamic array?

(xiii) You are given a rod of length n and an array of prices that contains prices of all pieces of size smaller than or equal to n. The problem is to find the maximum value obtainable by cutting up the rod and selling the pieces. Find the time complexity of the dynamic programming approach to solving this problem

(xiv) An approximation algorithm for a minimization problem guarantees a solution within an approximation ratio "r". If the cost of the optimal solution is C_{opt}, find cost C_{approx} of the solution found by the algorithm(in terms of r and C_{opt}) ?

(xv) Given a dynamic table that doubles in size when it reaches capacity, calculate the cost C_i of the **i^{th}** insertion operation

2. (a) Suppose we're doing a sequence of n operations (numbered 1, 2, 3,...) on a data structure in which the ith operations cost is as follows:

$$cost = \begin{cases} 1 \; if \; i \neq \; power \; of \; 2 \\ i \;\; if \; i = power \; of \; 2 \end{cases}$$

For example, the following table shows the costs for each of the first few operations:

operation number:	1	2	3	4	5	6	7	8	9 ...
cost:	1	2	1	4	1	1	1	8	1 ...

Use aggregate analysis to determine amortized cost per operation. [10]

(b) Prove that the average-case time complexity of Quicksort is O(n log n)

3. (a) Apply the 2-approximate algorithm for the Travelling Salesman Problem of the graph given in figure 1. [10]

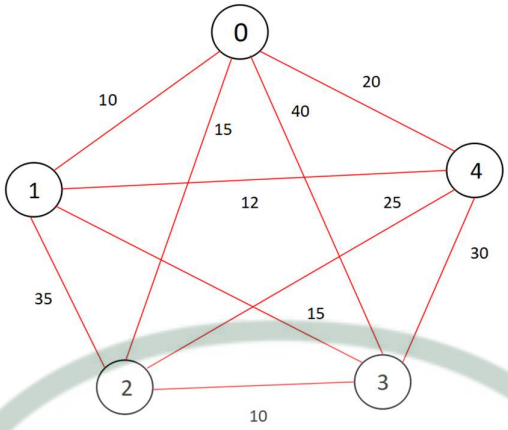

Figure 1: For TSP

(b) Trace the execution of Ford-Fulkerson algorithm to find the maximum flow of the graph given figure 2. [10]

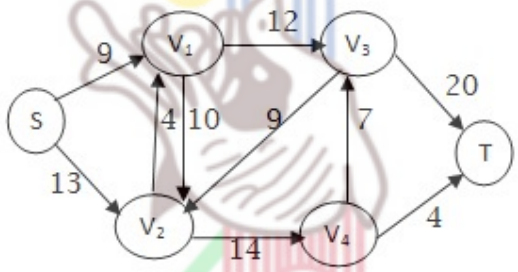

Figure 2: For Ford-Fulkerson algorithm

4. Solve the APSP problem using Floyd-Warshall's algorithm for the graph given figure 3. Clearly mention each steps for computing D (distance) and ∏(predecessor) matrices [20]

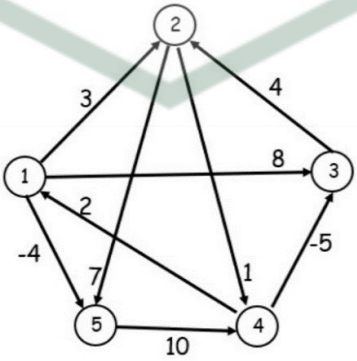

Figure 3: For Floyd-Warshall's algorithms

5. ***ANSWER ANY 4 QUESTIONS***

i. A neighbourhood has 5 intersections (A, B, C, D, E) and streets that need to be monitored using CCTV cameras. Each intersection can host a CCTV camera, which will cover all streets connected to that intersection. Place the minimum number of CCTV cameras at intersections in a neighborhood to monitor all streets, using an algorithm with O(n) complexity.
Neighbourhood Details:

- **Intersections (Vertices):** A, B, C, D, E
- **Streets (Edges):**
  - Street 1: A - B,
  - Street 2: A - C
  - Street 3: B - C
  - Street 4: B - D
  - Street 5: C - E

ii. Consider a small social network where people are connected if they are friends. The friendship connections are stored in an **n × n adjacency matrix** A, where A[i][j] = 1 if person i is a direct friend of person j, and A[i][j]=0 otherwise. The transitive closure of this adjacency matrix, denoted as T, represents **all possible indirect friendship connections**. That is, if there is a path of friends from person i to person j (either directly or through intermediaries), then T[i][j]=1; otherwise, T[i][j]=0

Given adjacency matrix A of the social network:

A=[0 1 0 0 1
1 0 1 0 0
0 1 0 1 0
0 0 1 0 1
1 0 0 1 0]

Generate friend recommendations for each user.

iii. Explain why random pivot selection in Quicksort ensures an average-case time complexity of (O(n log n)) and prevents worst-case behaviour in most cases.

iv. Explain the relationship between **P**, **NP**, and **NP-Complete** classes with examples.Prove that if an NP-Complete problem has a polynomial-time solution, then every problem in NP also has a polynomial-time solution (P = NP). Discuss the implications of this result.

v. Define what an approximation algorithm is and explain the **approximation ratio**. Why are approximation algorithms useful for NP-hard problems?